

IMPLEMENTASI ALGORITMA SIMULATED ANNEALING DALAM PENYELESAIAN MASALAH PENJADWALAN PRODUKSI JENIS *FLOWSHOP*

Aditya Ratnasari⁽¹⁾, Gloria Virginia⁽²⁾, Joko Purwadi⁽³⁾

Abstrak:

Penjadwalan produksi merupakan komponen yang sangat penting dalam sebuah proses produksi. Masalah penjadwalan yang sering ditemui dalam rantai produksi adalah masalah penjadwalan *flowshop* dan penjadwalan *jobshop*.

Sistem yang dibuat adalah tentang masalah penjadwalan *flowshop* dengan tujuan untuk mencari jadwal *job* yang mempunyai nilai *makespan* dan *total flow time* yang minimal menggunakan algoritma *Simulated Annealing*. Masukan berupa jumlah mesin, jumlah *job*, dan waktu proses setiap *job* pada masing-masing mesin. Proses yang dilakukan adalah mencari jadwal awal dengan algoritma SPT (*Shortest Processing Time*), kemudian menghitung nilai *makespan* dan *total flow time*, setelah itu pencarian jadwal yang paling optimal dengan menggunakan algoritma *Simulated Annealing*. Implementasi program dilakukan pada perusahaan UD. Wijaya Kusuma Brass.

Hasil penjadwalan dengan menggunakan algoritma *Simulated Annealing* telah diujicoba di perusahaan UD. Wijaya Kusuma Brass dan dibandingkan dengan hasil penjadwalan sebelum menggunakan *Simulated Annealing* pada perusahaan tersebut, dengan hasil lebih bagus menggunakan *Simulated Annealing*. Jadi, algoritma *Simulated Annealing* terbukti dapat digunakan untuk menyelesaikan masalah penjadwalan produksi jenis *flowshop* dengan tujuan minimasi *makespan* dan *total flow time*.

Kata Kunci: penjadwalan produksi, algoritma *Simulated Annealing*, *flowshop*

1. Pendahuluan

Pada dasarnya setiap perusahaan mempunyai tujuan yang sama yaitu mendapat keuntungan yang sebesar-besarnya dengan mengeluarkan biaya produksi seminimum mungkin. Hal tersebut dapat dicapai dengan cara menyelesaikan suatu pekerjaan tepat waktu dan memanfaatkan mesin secara optimal. Jumlah mesin yang terdapat dalam setiap usaha manufaktur berbeda-beda. Untuk memaksimalkan penggunaan keseluruhan mesin dan meminimalkan biaya produksi maka perlu dilakukan penjadwalan kerja yang tepat bagi setiap mesin.

Permasalahan yang dialami oleh perusahaan UD. Wijaya Kusuma Brass adalah melakukan penjadwalan dengan menggunakan intuisi dan pengalaman sehingga kadang-kadang terlambat dalam melakukan pengiriman barang. Dengan mengacu pada permasalahan tersebut, penulis mencoba menyelesaikannya dengan melakukan penjadwalan dengan tujuan untuk meminimalkan waktu penyelesaian sehingga perusahaan dapat melakukan pengiriman barang tepat waktu dan dapat meminimalkan total biaya yang dikeluarkan. Pada penelitian ini, algoritma *Simulated Annealing*

⁽¹⁾ Aditya Ratnasari, Mahasiswa Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana
Email : aditya_rtnsari@yahoo.com

⁽²⁾ Gloria Virginia, S.Kom., MAI., Dosen Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana

⁽³⁾ Joko Purwadi, S.Kom., M.Kom., Dosen Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana

digunakan untuk menyelesaikan kasus penjadwalan pada perusahaan UD. Wijaya Kusuma Brass.

2. Rumusan Masalah

Berdasarkan latar belakang masalah, maka dapat dirumuskan permasalahannya yaitu:

- a. Bagaimana hasil dari penerapan algoritma *Simulated Annealing* dalam menyelesaikan penjadwalan produksi jenis *flowshop* dengan tujuan minimasi *makespan* dan *total flow time*?
- b. Apakah dengan penjadwalan produksi akan meningkatkan lancarnya proses produksi pada perusahaan UD. Wijaya Kusuma Brass?

3. Batasan Masalah

Batasan masalah dari penelitian ini adalah:

- a. Sistem produksi adalah *flowshop* (sistem produksi yang setiap pekerjaan pasti melewati setiap mesin yang disusun berurutan).
- b. Tidak terdapat *pre-emption* (interupsi untuk mengerjakan suatu produk lain ditengah-tengah pengerjaan suatu produk).
- c. Setiap job memiliki *ready time* (waktu siap proses) yang sama dan bersifat *independent* terhadap yang lain yaitu *ready time* = 0.
- d. Mesin dianggap tidak pernah rusak dan selalu siap dioperasikan.
- e. Mesin dimungkinkan menganggur karena menunggu pekerjaan selanjutnya.
- f. Dalam sistem ini hanya ada jumlah maksimal banyak mesin adalah 20 mesin dan jumlah maksimal pekerjaan adalah 200 pekerjaan (dalam perusahaan UD. Wijaya Kusuma Brass terdapat 10 mesin dan dalam 1 minggu dapat menyelesaikan 100 pekerjaan, jadi asumsinya adalah perkembangan jumlah mesin dan pekerjaan pada perusahaan UD. Wijaya Kusuma Brass akan berkembang menjadi 2 kali lipat pada beberapa tahun ke depan).
- g. Sistem ini hanya bisa menyimpan dan membuka file dalam format *.txt.
- h. Sistem dalam pembuatannya menggunakan bahasa pemrograman Borland Delphi 7.0.

4. Dasar Teori

A. Penjadwalan

Penjadwalan adalah suatu proses pengalokasian sumber daya yang terbatas untuk melakukan beraneka ragam pekerjaan pada bidang manufaktur dengan mesin-mesin dan tenaga kerja sebagai sumber dayanya dan *job* sebagai tugas yang harus diselesaikan.

Pengertian penjadwalan menurut Michael Pinedo adalah proses pengambilan keputusan untuk menjalankan peran penting dalam bidang manufaktur dan sistem produksi seperti halnya dalam lingkungan pengolahan informasi.¹

Masalah penjadwalan muncul apabila pada saat yang sama terdapat sekumpulan pekerjaan yang harus diselesaikan dengan kendala yang dihadapi yaitu terbatasnya jumlah mesin atau fasilitas produksi yang tersedia. Salah satu usaha yang perlu dilakukan untuk mengatasi kendala tersebut adalah dengan menjadwalkan pekerjaan-pekerjaan tersebut sehingga dapat digunakan seoptimal mungkin. Penjadwalan yang baik akan memberikan dampak positif, yaitu rendahnya biaya produksi dan waktu pengiriman yang dapat meningkatkan kepuasan pelanggan.

Pengertian penjadwalan *flowshop* adalah sebuah sistem yang semua *job*-nya mempunyai aliran atau rute yang sama dan setiap *job* melewati setiap mesin hanya satu kali.

⁽¹⁾ Michael Pinedo, *Scheduling: Theory, Algorithms, and Systems*. New Jersey : Prentice Hall, Inc. 2nd Edition, 2002, hlm. 2.

b. Algoritma SPT (*Shortest Processing Time*)

Inti dari algoritma SPT adalah mencari waktu proses *job* yang terkecil dan mengurutkannya hingga yang terbesar. Pada tugas akhir ini, algoritma SPT digunakan untuk menentukan solusi awal dalam proses algoritma *Simulated Annealing*.

1) Algoritma *Simulated Annealing*

Simulated Annealing dikembangkan berdasarkan ide dari mekanisme perilaku pendinginan dan proses kristalisasi (*annealing*) material panas. Algoritma ini melakukan peningkatan iteratif untuk memperbaiki solusi yang dihasilkan teknik-teknik penjadwalan heuristik, dalam hal ini adalah sebuah solusi awal yang dibuat dengan teknik heuristik ataupun random, diiterasi secara berulang dengan metode *annealing* dengan menggunakan perturbasi lokal hingga tidak ada peningkatan lagi atau hingga jumlah iterasi yang diinginkan sudah dicapai.

Struktur algoritma *Simulated Annealing* secara umum adalah sebagai berikut:

- a) Evaluasi keadaan awal. Jika keadaan awal merupakan tujuan, maka pencarian berhasil dan KELUAR. Jika tidak demikian, lanjutkan dengan menetapkan keadaan awal sebagai kondisi sekarang.
- b) Inisialisasi BEST_SO_FAR untuk keadaan sekarang.
- c) Inisialisasi T sesuai dengan *annealing schedule*.
- d) Kerjakan hingga solusi ditemukan atau sudah tidak ada operator baru lagi akan diaplikasikan ke kondisi sekarang.
 1. Gunakan operator yang belum pernah digunakan tersebut untuk menghasilkan kondisi baru.
 2. Evaluasi kondisi yang baru dengan menghitung :
$$\Delta E = \text{nilai sekarang} - \text{nilai keadaan baru}$$
 - a. Jika kondisi baru merupakan tujuan, maka pencarian berhasil dan KELUAR.
 - b. Jika bukan tujuan, namun memiliki nilai yang lebih baik daripada kondisi sekarang, maka tetapkan kondisi baru sebagai kondisi sekarang. Demikian pula tetapkan BEST_SO_FAR untuk kondisi yang baru tadi.
 - c. Jika nilai kondisi baru tidak lebih baik dari kondisi sekarang, maka tetapkan kondisi baru sebagai kondisi sekarang dengan probabilitas :
$$p' = e^{-\Delta E/T}$$
Langkah ini biasanya dikerjakan dengan membangkitkan suatu bilangan random r pada range $[0, 1]$. Jika $r < p'$, maka perubahan kondisi baru menjadi kondisi sekarang diperbolehkan. Namun jika tidak demikian, maka tidak akan dikerjakan apapun.
 3. Perbaiki T sesuai dengan *annealing scheduling*.
- e) BEST_SO_FAR adalah jawaban yang dimaksudkan.

Parameter awal yang diperlukan:

- a) **Temperatur Awal**, merupakan penanda awal iterasi. Dimana nanti temperatur awal ini akan terus berkurang hingga mencapai temperatur akhir.
- b) **Temperatur Akhir**, merupakan batas akhir penanda iterasi sudah dapat dihentikan.
- c) **Faktor Reduksi Suhu**, merupakan angka yang digunakan untuk menurunkan suhu secara bertahap dan terkendali.

⁽¹⁾ Ronald G. Askin dan Jeffrey B. Goldberg, *Design and Analysis of Lean Production Systems*. United State of America : John Wiley & Sons, Inc., 2002, Hlm. 437.

⁽²⁾ Sri Kusumadewi dan Hari Purnomo, *Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik*. Yogyakarta : Graha Ilmu, 2005, Hlm. 177-178.

- d) **Angka replikasi**, merupakan angka yang menunjukkan berapa kali *loop* dalam harus dilakukan sebelum menurunkan suhu.

Penerapan algoritma *Simulated Annealing* pada permasalahan produksi jenis *flowshop*:

- Langkah awal yang harus dilakukan adalah dengan menentukan solusi awal yang didapatkan dengan algoritma SPT, dihitung nilai objektifnya yaitu MS (*makespan*) dan TF (*total flow time*).
- Menentukan parameter-parameter awal yang diperlukan, yaitu:
 - T_o = temperatur awal
 - T_a = temperatur akhir
 - k = konstanta faktor reduksi suhu, biasanya nilainya antara $0 < k < 1$
 - R = angka replikasi (angka yang menunjukkan berapa kali *loop* yang harus dilakukan sebelum menurunkan suhu)
- Mengambil dua angka random r_1 dan r_2 , antara 1 dan jumlah *job*, dimana r_1 tidak sama dengan r_2 .
- Menukarkan *job* pada posisi r_1 dengan *job* pada posisi r_2 .
- Menghitung nilai objektif jadwal baru ini, MS_{new} dan TF_{new} .
- Menghitung Δe dengan rumusan sebagai berikut:

$$e_1 = \left(\frac{MS - \min(MS; MS_{new})}{\min(MS; MS_{new})} \right) + \left(\frac{TF - \min(TF; TF_{new})}{\min(TF; TF_{new})} \right)$$

$$e_2 = \left(\frac{MS_{new} - \min(MS; MS_{new})}{\min(MS; MS_{new})} \right) + \left(\frac{TF_{new} - \min(TF; TF_{new})}{\min(TF; TF_{new})} \right)$$

$$\Delta e = e_2 - e_1$$

Keterangan:

- S = Solusi sebelumnya
- S_{new} = Solusi baru dari penukaran *job*
- MS = *makespan* solusi sebelumnya
- MS_{new} = *makespan* dari solusi baru
- TF = *total flow time* solusi sebelumnya
- TF_{new} = *total flow time* dari solusi baru
- $S(C_{j,m})$ = waktu penyelesaian dari *job* ke- j dalam m mesin untuk solusi sebelumnya
- $S_{new}(C_{j,m})$ = waktu penyelesaian dari *job* ke- j dalam m mesin untuk solusi baru
- $S(C_{n,m})$ = waktu penyelesaian dari n *job* dalam m mesin untuk solusi sebelumnya
- $S_{new}(C_{n,m})$ = waktu penyelesaian dari n *job* dalam m mesin untuk solusi baru
- MS = $S(C_{n,m})$
- TF = $\sum_{j=1}^n S(C_{j,m})$
- MS_{new} = $S_{new}(C_{n,m})$
- TF_{new} = $\sum_{j=1}^n S_{new}(C_{j,m})$

- g) Jika $\Delta e > 0$, maka hitung $p = \exp\left(-\frac{\Delta e}{T_o}\right)$. Ambil satu bilangan random pm antara

0 dan 1, jika $t_u > d$ maka jadwal baru diterima, jika tidak maka jadwal lama tetap dipertahankan.

Jika $\Delta > 0$, maka jadwal baru langsung diterima. Secara umum Δ memiliki pengertian selisih keunggulan jadwal baru terhadap nilai minimum yang dihasilkan dengan jadwal lama terhadap nilai minimum yang dihasilkan.

- h) Apabila kondisi replikasi lokal telah dipenuhi maka lakukan reduksi terhadap suhu dengan rumusan $T = \alpha \times T$.
- i) Kembali ke langkah tiga sampai kriteria penghentian iterasi terpenuhi.

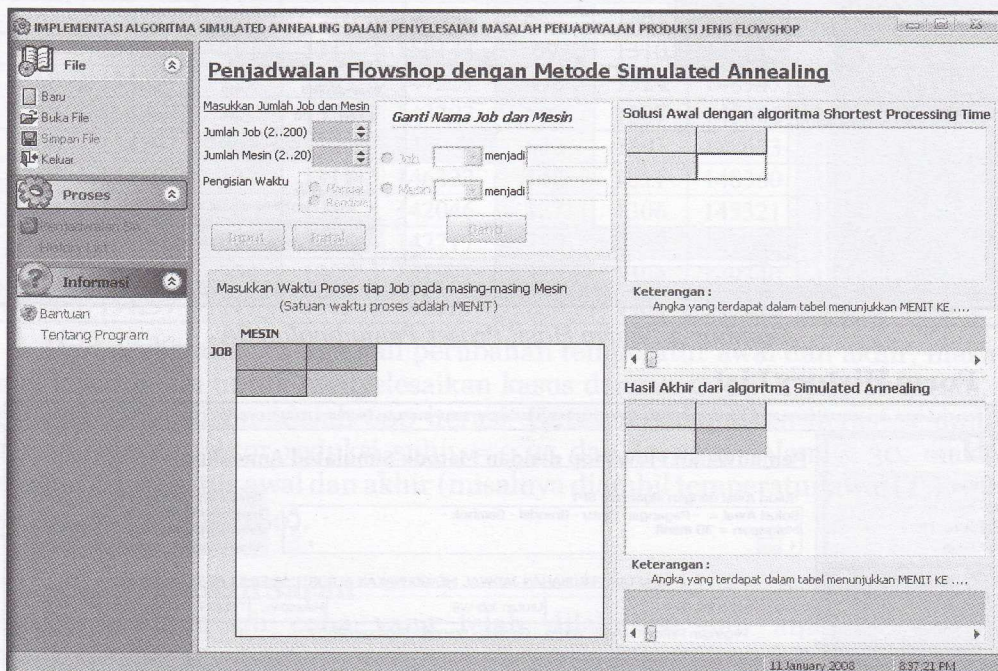
5. Implementasi

a. Form Tampilan Utama

Pada tampilan utama terdapat tiga menu bar yaitu File, Proses dan Informasi.

Masing-masing menu bar tersebut mempunyai sub menu, sebagai berikut:

- 1) **File**
 - a) Baru : digunakan untuk membuat file baru.
 - b) Buka File : digunakan untuk membuka file berekstensi *.txt, terbatas pada file yang dibuat menggunakan program ini.
 - c) Simpan File : digunakan untuk menyimpan file dengan ekstensi *.txt.
 - d) Keluar : digunakan untuk keluar dari program.



Gambar 1 : Form Tampilan Utama

2) Proses

- a) Penjadwalan SA : digunakan untuk melakukan proses penjadwalan dengan algoritma *Simulated Annealing*.
- b) History List : digunakan untuk melihat semua hasil dari proses penjadwalan *Simulated Annealing*.

3) Informasi

- a) Bantuan : digunakan untuk memberikan bantuan kepada pengguna ketika menemui kesulitan dalam menjalankan program.

b) Tentang Program : berisi informasi tentang pembuat program.

b. Form Proses Penjadwalan *Simulated Annealing*

Hasil dari proses penjadwalan SA tersebut diperlihatkan dalam bentuk tabel. Tabel yang pertama adalah tabel input waktu. Tabel yang kedua adalah tabel hasil dari solusi awal yang dilakukan dengan menggunakan algoritma SPT. Tabel yang ketiga adalah tabel hasil akhir setelah dilakukan proses penjadwalan SA. Jika pengguna ingin melihat hasil penjadwalan secara keseluruhan, maka dapat menekan menu History List.

Penjadwalan Flowshop dengan Metode Simulated Annealing

Masukkan Jumlah Job dan Mesin
 Jumlah Job (2..200) : 3
 Jumlah Mesin (2..20) : 4

Ganti Nama Job dan Mesin
 Job : [] menjadi []
 Mesin : [] menjadi []

Masukkan Waktu Proses tiap Job pada masing-masing Mesin
 (Satuan waktu proses adalah MENIT)

JOB	Mesin Lebur	Mesin Cetak	Mesin Kikir
Pegangan Pintu	7	1	1
Grendel	4	8	1
Gembok	3	2	9

Solusi Awal dengan algoritma Shortest Processing Time

Job/Mesin	Mesin Lebur	Mesin Lebur	Mesin Cetak	Mesin Cetak	N
Pegangan Pintu	7	7	8	8	8
Grendel	7	11	11	19	1
Gembok	11	14	19	21	2

Hasil Akhir dari algoritma Simulated Annealing

Job/Mesin	Mesin Lebur	Mesin Lebur	Mesin Cetak	Mesin Cetak	N
Gembok	0	3	3	5	5
Pegangan Pintu	3	10	10	11	1
Grendel	10	14	14	22	2

Gambar 2 : Form Hasil Proses Penjadwalan SA

c. Form History List

Penjadwalan Flowshop dengan Metode Simulated Annealing

Solusi Awal dengan algoritma SPT
 Solusi Awal = - Pegangan Pintu - Grendel - Gembok -
 Makespan = 30 menit

Temperatur Awal = 1000
 Temperatur Akhir = 70
 Konstanta Faktor Reduksi Suhu = 0.95
 Angka Replikasi (Looping Dalam) = 5

DAFTAR PERUBAHAN JADWAL MENGGUNAKAN ALGORITMA SIMULATED ANNEALING

Iterasi	Urutan Job CS	Urutan Job WS	Makespan	Total FT	Temperatur	R	e1
1	- Pegangan Pintu - Grendel - Gembok	- Gembok - Grendel - Pegangan Pin	17	47	1000	1	1.02
2	- Gembok - Grendel - Pegangan Pin	- Grendel - Gembok - Pegangan Pin	24	60	1000	2	0
3	- Grendel - Gembok - Pegangan Pin	- Grendel - Pegangan Pintu - Gembok	25	52	1000	3	0.15
4	- Grendel - Pegangan Pintu - Gembok	- Grendel - Gembok - Pegangan Pin	24	60	1000	4	0.04
5	- Grendel - Gembok - Pegangan Pin	- Grendel - Pegangan Pintu - Gembok	25	52	1000	5	0.15
6	- Grendel - Pegangan Pintu - Gembok	- Grendel - Gembok - Pegangan Pin	24	60	950	1	0.04
7	- Grendel - Gembok - Pegangan Pin	- Pegangan Pintu - Gembok - Grendel	23	53	950	2	0.17
8	- Pegangan Pintu - Gembok - Grendel	- Gembok - Pegangan Pintu - Grendel	23	52	950	3	0.01
9	- Gembok - Pegangan Pintu - Grendel	- Grendel - Pegangan Pintu - Gembok	25	52	950	4	0
10	- Grendel - Pegangan Pintu - Gembok	- Pegangan Pintu - Grendel - Gembok	20	50	950	5	0

Jumlah Job : 3
 Jumlah Mesin : 3

Jadwal Terbaik terdapat pada Iterasi ke : 1

PRINT KEMBALI

Gambar 3 : Form History List

Pada History List, pengguna dapat melihat hasil solusi awal dengan SPT, informasi temperatur awal, temperatur akhir, konstanta faktor reduksi suhu, angka replikasi (looping dalam), informasi hasil iterasi dengan algoritma *Simulated Annealing* serta informasi jadwal terbaik yang dihasilkan. Tombol Kembali digunakan untuk kembali ke gambar 2.

d. Analisis Perubahan Temperatur

Pada analisis ini akan digunakan data 200 job 20 mesin dengan waktu proses secara *random*. Parameter awal yang digunakan pada percobaan kali ini adalah

Temperatur antara 1-1000

k (konstanta faktor reduksi suhu) = 0.95

R (jumlah looping dalam) = 30

Tabel 1 : Tabel Percobaan 200 Job dan 20 Mesin

No	Hasil Sebelum SA		Hasil SA dengan $T_0 = 500$ dan $T_a = 300$ 300 Iterasi			Hasil SA dengan $T_0 = 500$ dan $T_a = 180$ 600 Iterasi			Hasil SA dengan $T_0 = 1000$ dan $T_a = 70$ 1560 Iterasi		
	Make span	Total Flow Time	Iterasi ke-	Make span	Total Flow Time	Iterasi ke-	Make span	Total Flow Time	Iterasi ke-	Make span	Total Flow Time
1	1368	154297	198	1310	147534	556	1301	147976	1560	1304	147916
2	1368	154297	297	1313	150144	69	1310	145819	1394	1309	148043
3	1368	154297	122	1303	147221	436	1324	148947	1353	1316	148931
4	1368	154297	48	1325	143203	500	1307	149606	1210	1324	147497
5	1368	154297	261	1303	146934	600	1303	148673	1488	1318	149109
6	1368	154297	133	1311	146322	542	1311	148360	1507	1332	148914
7	1368	154297	8	1325	142046	127	1306	145321	1474	1305	145861
8	1368	154297	28	1317	142717	595	1299	147740	1067	1326	148684
9	1368	154297	234	1307	147591	379	1304	149340	1414	1306	146972
10	1368	154297	147	1311	146927	220	1304	147292	969	1306	146275

Dari perbandingan tiga kali perubahan temperatur awal dan akhir, maka dapat diambil kesimpulan untuk menyelesaikan kasus dengan 200 job dan 20 mesin, iterasi yang direkomendasikan adalah 600 iterasi. Untuk mendapatkan iterasi sejumlah 600 dengan konstanta faktor reduksi suhu = 0.95 dan *looping* dalam = 30, maka dapat ditentukan temperatur awal dan akhir (misalnya diambil temperatur awal (T_0) = 500 dan temperatur akhir (T_a) = 180).

6. Kesimpulan dan Saran

Berdasarkan uji coba yang telah dilakukan dan analisis sistem, serta memperhatikan seluruh proses yang terjadi, maka dapat diperoleh kesimpulan sebagai berikut:

- Masalah penjadwalan produksi *flowshop* terbukti dapat diselesaikan menggunakan algoritma *Simulated Annealing* dengan tujuan meminimalkan *makespan* dan *total flow time*.
- Tujuan meminimalkan *makespan* dan *total flow time* dapat dicapai dengan memberikan parameter awal yang tepat.
- Hasil akhir yang didapat dari algoritma *Simulated Annealing* selalu berubah dan belum tentu merupakan hasil yang terbaik karena algoritma *Simulated Annealing* mengambil solusi tetangga secara *random*.
- Implementasi program pada perusahaan UD. Wijaya Kusuma Brass dapat antu kelancaran proses produksi karena waktu penyelesaian menjadi lebih minimum.

membantu kelancaran proses produksi karena waktu penyelesaian menjadi lebih minimum.

Beberapa saran untuk pengembangan sistem lebih lanjut adalah:

- Tampilan pada sistem dapat dikembangkan supaya lebih baik lagi, dengan memberikan suatu grafik yang memberi informasi perubahan *makespan* dan *total flow time* selama iterasi berlangsung, sehingga dapat memberikan informasi tentang *makespan* dan *total flow time* minimum yang pernah dicapai sistem dalam iterasi.
- Membuat sistem yang dapat meminimalkan waktu menganggur mesin.
- Membandingkan algoritma *Simulated Annealing* dengan algoritma lainnya, sehingga lebih mengetahui performa dari algoritma *Simulated Annealing*.
- Proses dapat berhenti jika solusi yang didapatkan sudah minimal meskipun temperatur akhir belum dicapai.

No	Iterasi	Temperatur Awal (T _{awal}) = 500				Temperatur Akhir (T _{akhir}) = 100			
		Makespan	Total Flow Time	Makespan	Total Flow Time	Makespan	Total Flow Time	Makespan	Total Flow Time
1	1	1308	124397	1308	124397	1308	124397	1308	124397
2	2	1308	124397	1308	124397	1308	124397	1308	124397
3	3	1308	124397	1308	124397	1308	124397	1308	124397
4	4	1308	124397	1308	124397	1308	124397	1308	124397
5	5	1308	124397	1308	124397	1308	124397	1308	124397
6	6	1308	124397	1308	124397	1308	124397	1308	124397
7	7	1308	124397	1308	124397	1308	124397	1308	124397
8	8	1308	124397	1308	124397	1308	124397	1308	124397
9	9	1308	124397	1308	124397	1308	124397	1308	124397
10	10	1308	124397	1308	124397	1308	124397	1308	124397

Dari perbandingan ini kita dapat melihat bahwa dengan menggunakan algoritma *Simulated Annealing* dengan temperatur awal 500 dan temperatur akhir 100, maka dapat diperoleh solusi yang lebih baik dibandingkan dengan algoritma lain yang digunakan. Hal ini dapat dilihat dari nilai *makespan* dan *total flow time* yang lebih kecil dibandingkan dengan algoritma lain yang digunakan.

4. Kesimpulan dan Saran
Berdasarkan uraian di atas yang telah dilakukan dan analisis yang telah dilakukan, maka dapat disimpulkan bahwa sistem yang dikembangkan dapat membantu kelancaran proses produksi karena waktu penyelesaian menjadi lebih minimum.

- Membuat tampilan yang lebih baik lagi, dengan memberikan suatu grafik yang memberi informasi perubahan *makespan* dan *total flow time* selama iterasi berlangsung, sehingga dapat memberikan informasi tentang *makespan* dan *total flow time* minimum yang pernah dicapai sistem dalam iterasi.
- Membuat sistem yang dapat meminimalkan waktu menganggur mesin.
- Membandingkan algoritma *Simulated Annealing* dengan algoritma lainnya, sehingga lebih mengetahui performa dari algoritma *Simulated Annealing*.
- Proses dapat berhenti jika solusi yang didapatkan sudah minimal meskipun temperatur akhir belum dicapai.